The Development of Web Security Scanner Based on XSS and SQL Injection Method

Ibnu Gunawan
Petra Christian University
Siwalan kerto 121 - 131
Surabaya
+62312983456
ibnu@petra.ac.id

Agustinus Noertjahyana
Petra Christian University
Siwalan kerto 121 - 131
Surabaya
+62318439040
agust@petra.ac.id

Deddie Tjahjono
Petra Christian University
Siwalan kerto 121 - 131
Surabaya
+62318439040
M26407051@john.petra.ac.id

ABSTRACT

Nowaday, there is so many vulnerabilities in web application layer. This is because of security issues that are often overlooked by a web developer when creating a website. In fact, caused by the presence of vulnerabilities on a website, a hacker can do a variety of activities that destroy of website. Adverse events that can be done by a hacker includes changing the web page (defacing), obtain sensitive information, even taking over control of the website system. To help overcome these problems, we make an application to detect vulnerabilities that exist on a website.

The process is started by crawling to get the entire link from the target website. Followed by attacking the process that is useful to attempt an attack on a link that has the potential security hole. The application will then continue in the process of reporting where the application would create a vulnerability report on the website. This application was built using Microsoft Visual C # 2010.

Based on the results of tests made on this application, it can be concluded that the application can detect vulnerabilities in the website and report any form of link that has a security hole on the website.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: *Unauthorized access*

General Terms

Security

Keywords

Web, application, security, scanner, xss, sql, injection

1. INTRODUCTION

Security of the application site should be a priority for a web administrator and web developer. But generally the manufacturer's website only give priority to the design and what topics are provided in order to attract as many visitors. Website security is usually placed on the nth order. Though the website application security is the most important because of the presence of vulnerabilities on the website then the website will be attacked by hackers [1].

Based on the 2009 report from WASC (Web Application Security Consortium), an organization that examines the field of web application security, attacks on the application site is increasing every year. Which, from his report said that until the end of the epidemic in 2009, 87% of the total existing websites still have gaps that can be fatal to the application site [2]. Of the report, said that the security hole is in the most XSS (Cross-Site Scripting) as much as 39%, followed by 32% Information Leakage, SQL Injection and as much as 7%.

The impact of the gap is not only detrimental to the developer but also detrimental to the user. The number of attacks in this layer due to the application site is very easy to attack because in general the application site has many weaknesses and easily exploited.

Therefore, to help web developers in tackling this problem, we need Web Application Security Scanner for detecting a variety of security holes in the website and produce a report in the form of a report containing an overview of the security holes in the website automatically [3].

2. PRELIMINARIES

In this section we briefly introduce about web application layer, its security, and web application security scanner. Then we moved to it's method especially the sql injection and xss.

2.1 Website Application Layer

Website or application layer can also be called the application layer is a layer of websites that act as primary liaison with the website users around the world [4]. There is also a database that contains highly sensitive information like credit card number, name, address, date of birth, financial records, trade secrets, medical records, and many other important information.

This layer is usually the main attack for hackers who have a variety of purposes, such as information theft, damage the website, or even take over control of the website. This layer was attacked because it is the weakest layer than other layers. Along with the development of technology, the automatic will indirectly create a more complex application layer. Where this is going to make so many of them the possibility of bugs and loopholes that can be exploited by hackers as a way into the system.

From the report WASC (Web Application Security Consortium), said that currently 75% of cyber attacks in the world begins at the application layer and the website can be fatal to the website [2].

2.2 Website Application Security

Web application security is a range of measures taken to protect the application layer on the website of the hacker attacks that can cause a variety of losses for individuals and companies the website owner[5].

2.3 Website Application Security Scanner

Web Application Security Scanner is the software that automatically search for vulnerabilities that exist on the website. This software does not access the source code in carrying out its action, the means used this software to detect security loopholes that exist is by Black Box [3]. Another name for this kind of application is a Web Application Vulnerability Scanner

These applications are generally divided into three stages in carrying out its action, namely Crawling Component, Component Attack, and Analysis Modules [3]:

- Crawling Component or a term popular with the Web Crawler, where the website will index the links on the website. This step can be performed by various methods, depending on the needs of application users.
- Attack Component, in which the application will start automatically attempted assault on a link that has been indexed.
- Analysis Modules in which the application will evaluate the responses provided by the website and create reports of website security picture

2.4 Web Crawler

Web crawlers are also commonly known as the Spider Web. Where this method has a duty to collect all the information in our website. Work performed by the Web Crawler is done automatically by the record of each link on the website pages you visit and then visit these links one by one. Its implementation, there are various methods of web crawlers that are used as needed [6]

The methods commonly used in Web Crawler is:

- BFS (Breadth First Search)
  Search based on the breadth of available information website, which on its use as a storage utilizing URL Queue
- DFS (Depth First Search)
  Search depth of information available on the website, which on its use as a storage URL Stack harness

In short, a Web crawler process generally begins by providing a set of initial seed URL as the search into a queue. Priority criteria can be applied to reorder the list of URLs in the queue. The next crawler to download web pages by URL is retrieved from the queue. Once deposited into the collection, obtained parsed pages (parsed) to be extracted out-going unvisited link and then inserted into the queue. Pick-up process continues until the web page URL queue is empty or if the stop condition is met. Figure 2.1. shows a web crawler.

Input: $Seed = \{u_1, u_2, \ldots, u_n\}$ daftar URL awal
$URL\_Pool \leftarrow Seed$
$Visited \leftarrow \emptyset$, URL yang telah di kunjungi
while $URL\_Pool \neq \emptyset$
  $u \leftarrow$ **Select** (URL_Pool, Kriteria Pemilihan)
  $p \leftarrow$ **Download** (u)
  $Visitied \leftarrow Visited \cup u$
  $out\_link \leftarrow$ **Extract_Outgoing_Link** (p)
  for each $q \in out\_link$
    if $(q \notin Visited)$ and $(q \notin URL\_Pool)$
      $URL\_Pool \leftarrow URL\_Pool \cup q$
    end if
  end for
end while

Figure 2.1 Web Crawler Genetic Algorithm

2.5 Regular Expression

Regular Expression or more often called Regex is a technique used to match a text string, such as particular characters, words, or patterns of characters. Regex has two main functions, ie search and replace, find a certain pattern in the text and then switch to another patter [8]. Some common patterns used in the regex shown in table 1:

Table 1. Table captions should be placed above the table

Symbol	meaning
*	Replace character to infinte
+	Replace one character to infinte
?	Replace character 1 and 0
^	Search for a word that begins by pattern
$	Search for a word that ends the pattern
|	Give a choice
()	Make sub pattern

2.6 SQL Injection

SQL Injection is a technique that utilizes SQL query writing errors on a website so that a hacker could add some SQL statements to the 'query' by manipulating data input into the application. So that the database server to generate an invalid SQL query[7]. On the reality, SQL Injection is a proven one of the best techniques that often paralyze the target. With this technique the attacker can log into the system without having an account. Figure 2.2 show a SQL Injection syntax

```
http://10.252.108.232/web1/index.php?option=product.php&status
=1;update barang set harga = 50 where barangID=9;
```

Figure 2.2 sql injection via URL

2.7 XSS (Cross Site Scripting)

Cross Site Scripting is a type of attack where the method used is to inject Javascript into a website. Attacks of this type is usually underestimated because in most cases have an impact on the client or the so-called Client-Side Script. But in fact this kind of fatal attack[9], because an attacker could potentially do the following:

- Users can inadvertently run a script that has been inserted by the attacker and open the content according to the script.
- The attacker can take over from the user's active session before the session expired. Where the impact is the attacker can get into user accounts without having to make the login process.
- Attackers can connect users automatically to the server designated by the attacker.

to know that the website has a XSS vulnerability we can use method of Request / Response Match[10]. Where this method is trying to insert XSS code in the URL and make requests to the webserver. When the webserver responds in the form of content that contains XSS code, it can be said that the website has a security gap in this field. Figure 2.3 show a flowchart of Request / Response Match method

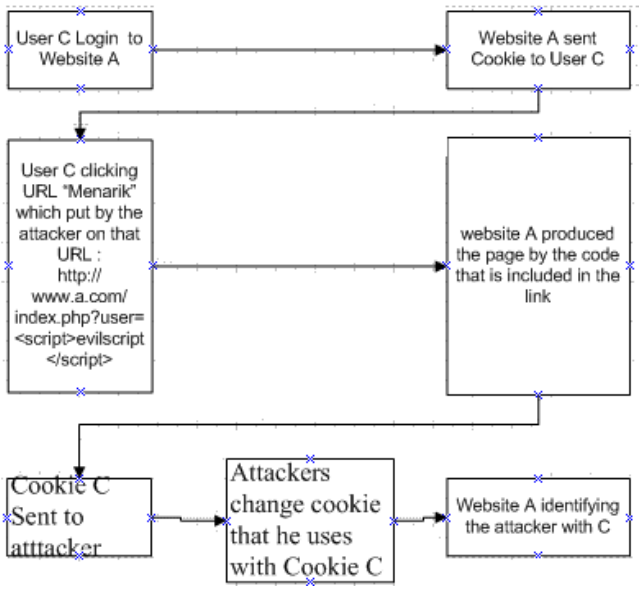

Figure 2.3 flowchart of request / response match method

If the XSS injection performed on a variable that simply pass a parameter without saving it in the database, then the results are only temporary (temporary). But if this weakness is found in the Guest Book, Shout Book, Forum, Blog, and the like and do XSS attack, the result of such attacks would be permanent because the injected script is stored in the database.

3. SYSTEM DESIGN

The software has been developed is consists of several stages. Start by crawling process to get a website structure, conduct attacks experiments on the pages that have the potential to have security holes, and displays the results of an overview report of website vulnerabilities. Figure 3.1 show the main flowchart.

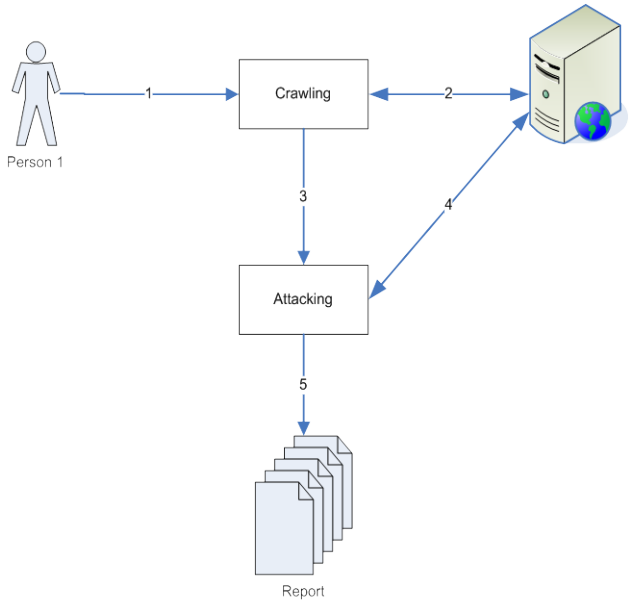

Figure 3.1 main system flowchart

3.1 Threading Process

At the beginning of each function to be executed, held the settings thread, where the software does is to ask the user how many threads and timeout to be used in the process. Timeout setting is used as the reference length of the website provides a response. While the threads are useful to accelerate the setting process of a function. Figure 3.2 show the main threading process

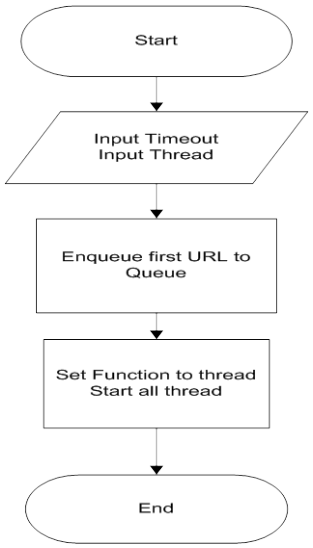

Figure 3.2 main threading process

3.2 Crawling process

Crawling process is a process where the software will scan the results to the response given from the server to get the links on the website. Where the process is done recursively, allowing the software to index the entire link from the website.

The system provides additional features that allow users to see the running processes Crawling, saw logs and structure of the website is being crawled, and see the link is ignored. Figure 3.3 show the main flowchart for the crawling process

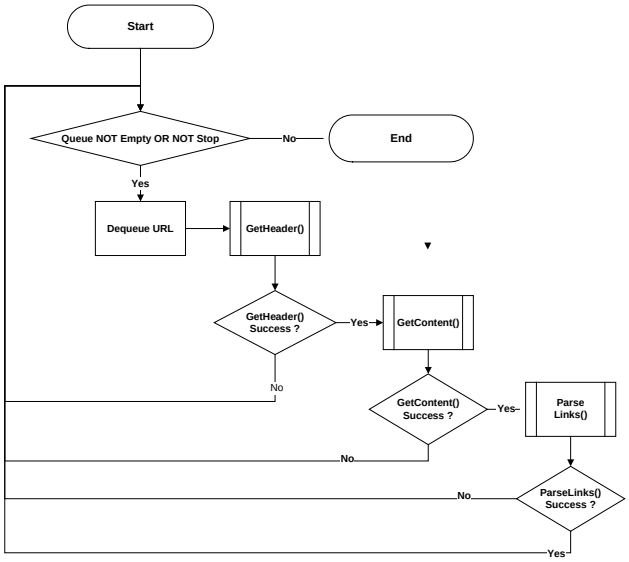

Figure 3.3 main flowchart crawling process

3.3 Attacking Proccess

Attacking process is a process in which the experiments will be carried out attacks on the link that has been obtained from the crawling process. From the results of such an attack would be detected if the link is being attacked have security holes in certain areas. The results of this process will be sent to the Report to create reports on website security picture

At this process, the software will ask the user what type of experiments that will be attempted attack on the target website. Where the trial will be conducted in accordance with user's choice. Figure 3.4 show the main process of attacking process

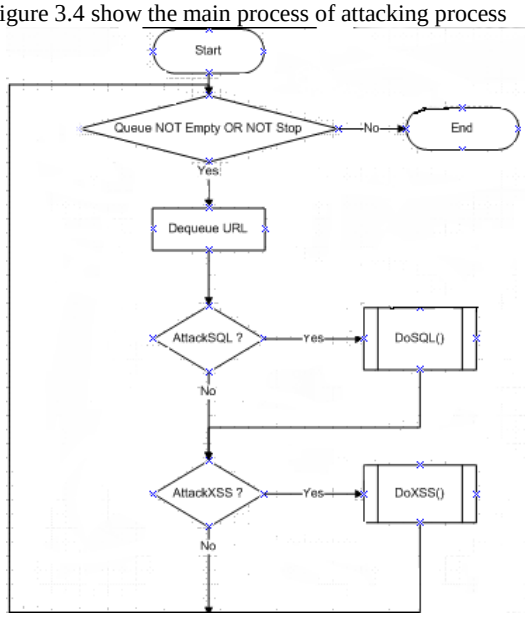

Figure 3.4 attacking process main flowchart

4. IMPLEMENTATION

Web application is made using the programming language C # using Microsoft Visual C # 2010 as an Integrated Development Environment (IDE) from the C # language. The reason the use of the language C # is because the ability of C # which is known to OOP (Object Oriented Programming) is fairly easy to apply and the language is quite easy to implement, as well as the compiler Microsoft Visual C # are very supportive of designing GUI (Graphical User Interface) and very flexible to use.

Whole process is made using a namespace that already provided by Microsoft Visual C # 2010. This is because C # does not have its own class library, so the use of class library in C # using a class library that is used in Visual Basic and Visual C + +. The namespace is used as follows:

- System.Threading, used for setting the thread that will be used to accelerate the course of these functions in the application.
- System.Text.RegularExpressions, used to set the regular expression pattern that will be used to detect the link on the website content
- System.IO, used for setting the Save and Load data from files that are used in applications. Including storage of the final report as a result of the application.
- System.Net, is used to request a link to the web server settings

Figure 4.1 show the class structure that has been made for this web application

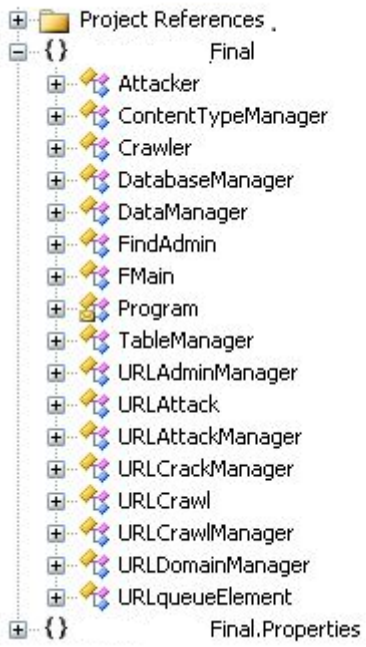

Figure 4.1 class structure diagram

The main classes used in the application is:

- Crawler, a class that contains functions to perform the process of crawling on the target website.
- URLCrawlManager, a class that contains functions for setting the course of the process of crawling, where the class is setting the thread to do the crawling.
- URLCrawl, a class that is used to store all data generated by the crawling process

- URLQueueElement, a class that contains the data to be inserted into the queue used to process the application.
- The attacker, a class that contains functions to perform the process of attacking the target website.
- URLAttackManager, a class that contains functions for setting the course of the process of attacking, in which the class is setting the thread to do the attacking.
- URLAttack, a class that is used to store all data generated by the process of attacking
- URLDomainManager, a class that is used to store the domain obtained during the application process.
- URLContentTypeManager, a class that is used to store the content type of link that obtained during the process of crawling on the application.
- FindAdmin, a class that contains functions to perform the search process on the target website admin page.
- URLAdminManager, a class that contains functions for setting the course of the search process admin page, where the class is setting the thread to the admin page to do searches.
- URLCrackManager, a class that contains functions to perform the process Crack Data on the target website.
- DatabaseManager, a class whose function is to store the entire database is available on the Crack Data. TableManager, a class whose function is to store the entire database structure is successfully obtained.
- DataManager, a class whose function is to store all data from a database which is found in the Crack Data

5. PROGRAM TESTING

A first step the use of a user's system will automatically be transferred to the tab Crawl, where the tab is a link the user to enter input or website name to be researched vulnerabilities. The initial steps to be performed is the process of crawling. Crawling is the process of the initial stage where the application will try to make repeated requests to get all the links on the website. Users can also set the number of threads used options at the time of the Crawling.

Figure 5.1 show menu of the main screen of our application, figure 5.2 show the option for proxy and user agent, figure 5.3 show the crawling process. Then we continuing our attack using xss which is setting is shown in figure 5.4 after that we continuing testing using sql injection which is setting is shown in figure 5.5. the result is shown in figure 5.6, 5.7 and 5.8 for reporting

Based on experiment that have been done, we can see that xss and sql injection can caused enough trouble for non hacker prepared site.

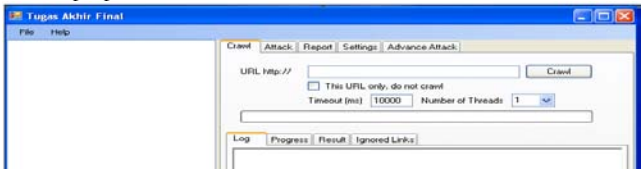

Figure 5.1 Web App Vulnerability scanner main screen

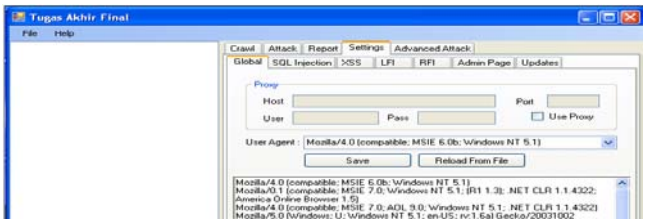

Figure 5.2 the option for proxy and user agent

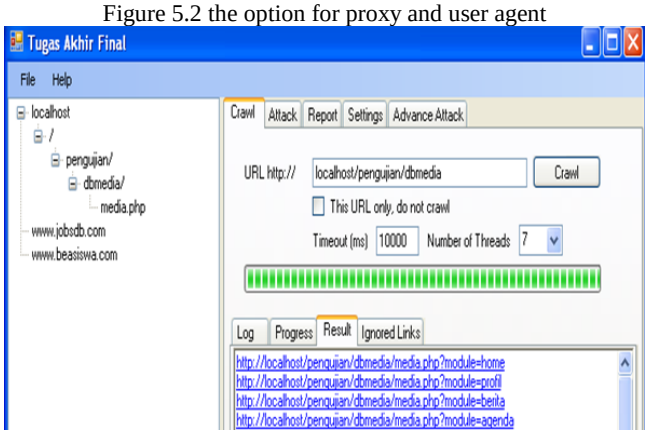

Figure 5.3 crawling process

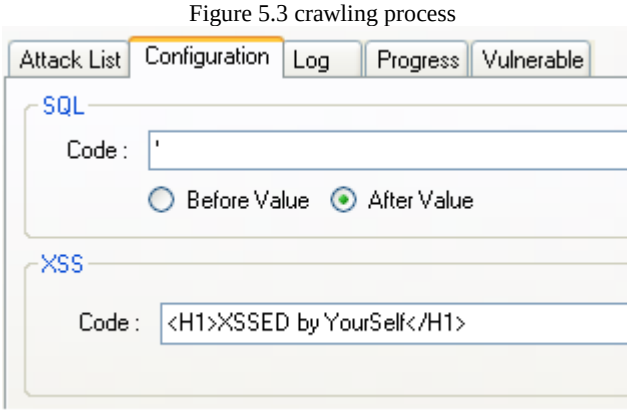

Figure 5.4 xss setting

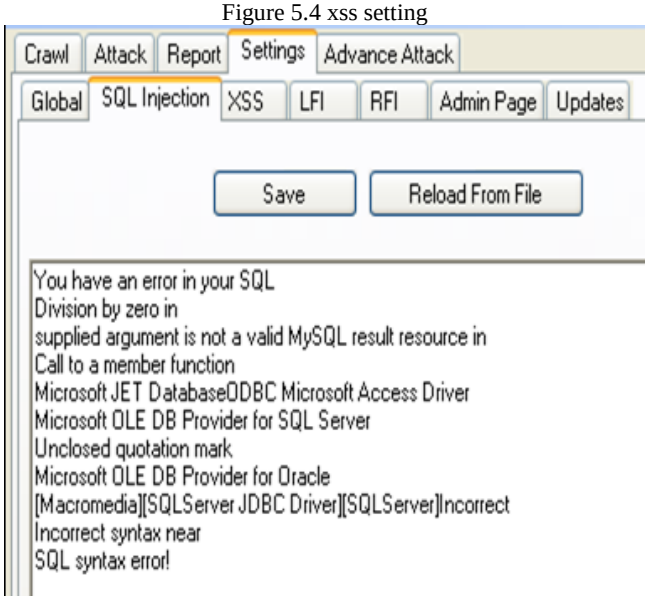

Figure 5.5 sql injection setting

Figure 5.6 sql injection result

Figure 5.7 XSS attack result

Figure 5.8 attack result report

6. CONCLUSION

Based on systems that have been developed and test results that have been done, we can conclude some of the following:

- The crawling process can affect the outcome of the report is generated, this is because the link that will be processed on the next stage (Attacking) is a link that has been obtained in the process of crawling
- The test can be seen that application of the flaw could be used as link testing. It has been proved by testing manually obtained from the link.
- The first test to prove the security hole in the areas of SQL Injection can be exploited by attackers to get the entire database structure and data-sensitive data from target website.
- In the first test also seen the gap XSS flaws, where the gap is an attacker can inject javascript. It can be dangerous because an attacker can modify it for various uses javascript attacks.

7. REFERENCES

[1] Huang, Yao-Wen et al. (2004). *Non-detrimental web application security scanning*, ISSRE, pp.219-230, 15th International Symposium on Software Reliability Engineering (ISSRE'04).

[2] Orloff, J. (2009). *Web application security: Testing for vulnerabilities.* New York: Sequoia Media Services. Inc.

[3] Kals, Stefan, et al. (2006). *Secubat: A web vulnerability scanner.* Technical Security University of Vienna

[4] Black, Paul et al. (2008). *Software assurance tools:Web application security scanner functional specification version 1.0.* National Insitute of Standards and Technology. United States.

[5] Stasiak, K. (2002). *Web application security.* Ohio: SecureState, Inc.

[6] Widiantoro, Dwi, H. (2006). *Survey arah penelitian, pengembangan, dan penerapan penjelajah situs web.* Proceeding of International Conference on Instrumentation, Communication and Information Technology, Insitut Teknologi Bandung.

[7] Sunyoto, A. (September 2004). Metode penyerangan website menggunakan SQL Injection.*Jurnal DASI* 5 (3), Retrieved 8 November 2010 from http://journal.amikom.ac.id/index.php/informatika/article/viewArticle/124

[8] Cho, J. (2002). *A fast regular expression indexing engine.* Proceeding of 18th International Conference on Data Engineering. University of California.

[9] Saha, S. (2009). *Consideration points: Detecting cross site scripting. International Journal of Computer Science and Computer Security (IJCSIS)*, Hanyang University.

[10] Johns, Martin et al. (2008). *XSSDS: Server-side detection of cross scripting attacks*, 24th Annual Computer Security Applications Conference (ACSAC '08), pp. 335 - 344, IEEE Computer Society.